

### 3.1.4 缓冲区

应用程序与通道的交互主要是进行数据的读取和写入。为了完成 NIO 的非阻塞读写操作，NIO 为大家准备了第三个重要的组件——Buffer。所谓通道的读取，就是将数据从通道读取到缓冲区中；所谓通道的写入，就是将数据从缓冲区写入通道中。缓冲区的使用是面向流进行读写操作的 OIO 所没有的，也是 NIO 非阻塞的重要前提和基础之一。

接下来笔者从缓冲区开始为大家详细介绍 NIO 的三大核心组件。

## 3.2 详解 NIO Buffer 类及其属性

NIO 的 Buffer 本质上是一个内存块，既可以写入数据，也可以从中读取数据。Java NIO 中代表缓冲区的 Buffer 类是一个抽象类，位于 `java.nio` 包中。

NIO 的 Buffer 内部是一个内存块（数组），与普通的内存块（Java 数组）不同的是：NIO Buffer 对象提供了一组比较有效的方法，用来进行写入和读取的交替访问。

#### 说明

Buffer 类是一个非线性安全类。

### 3.2.1 Buffer 类

Buffer 类是一个抽象类，对应于 Java 的主要数据类型。在 NIO 中，有 8 种缓冲区类，分别是 `ByteBuffer`、`CharBuffer`、`DoubleBuffer`、`FloatBuffer`、`IntBuffer`、`LongBuffer`、`ShortBuffer`、`MappedByteBuffer`。前 7 种 Buffer 类型覆盖了能在 IO 中传输的所有 Java 基本数据类型，第 8 种类型是一种专门用于内存映射的 `ByteBuffer` 类型。不同的 Buffer 子类可以操作的数据类型能够通过名称进行判断，比如 `IntBuffer` 只能操作 `Integer` 类型的对象。

实际上，使用最多的是 `ByteBuffer`（二进制字节缓冲区）类型，后面的章节会看到它的具体使用。

### 3.2.2 Buffer 类的重要属性

Buffer 的子类会拥有一块内存，作为数据的读写缓冲区，但是读写缓冲区并没有定义在 Buffer 基类中，而是定义在具体的子类中。例如，`ByteBuffer` 子类就拥有一个 `byte[]` 类型的数组成员 `final byte[] hb`，可以作为自己的读写缓冲区，数组的元素类型与 Buffer 子类的操作类型相对应。

#### 说明

在本书的上一个版本中，这里的内容为：Buffer 内部有一个 `byte[]` 类型的数组作为数据的读写缓冲区。乍看上去没有什么错误，实际上那个结论是错误的。具体原因是作为读写缓冲区的数组，并没有定义在 Buffer 类中，而是定义在各具体子类中。感谢社群小伙伴@炬，是他发现了这个比较隐蔽的编写错误。